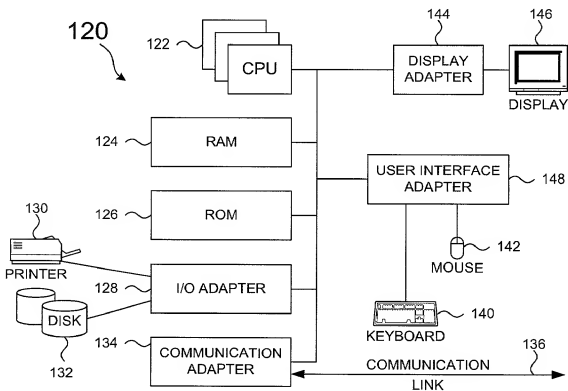
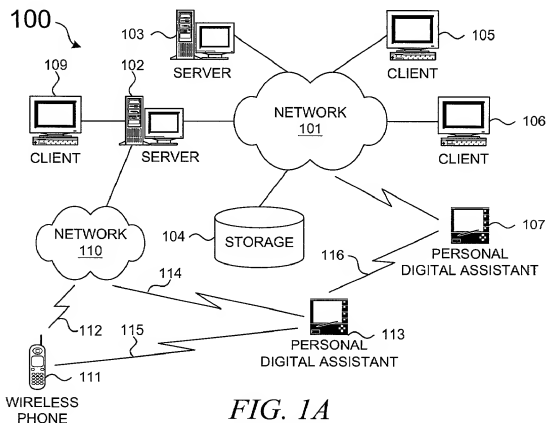


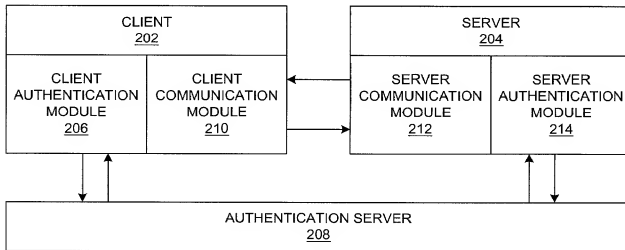
System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

1/10

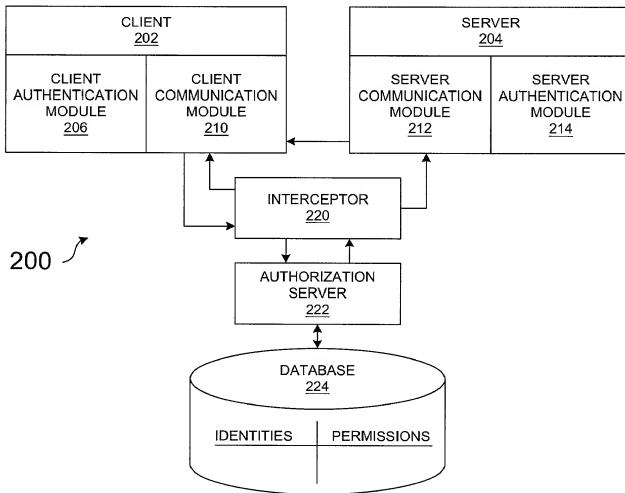


System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

2/10



200 *FIG. 2A*
(PRIOR ART)



200 *FIG. 2B*
(PRIOR ART)

System and process for enhancing method calls of special purpose object-oriented
programming languages to have security attributes for access control

3/10

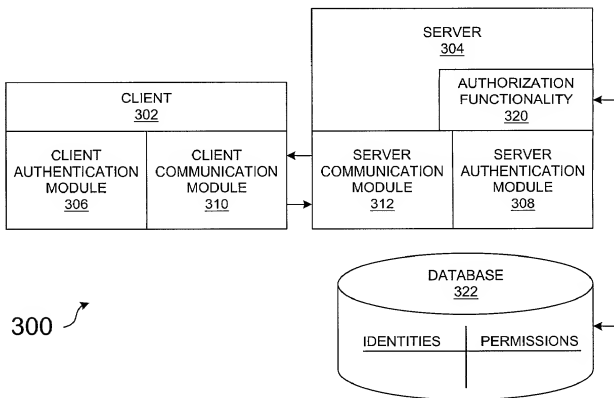


FIG. 3

System and process for enhancing method calls of special purpose object-oriented
programming languages to have security attributes for access control

4/10

402 {
Class public ClientClassA {
 Method A {
 try {
 ServerClassM.M(); // Might throw an authorization exception.
 // Try block successfully executes when
 // invoked by entity with proper authority,
 // such as BankManager
 }
 catch (AuthException e)
 {
 // Codepath not reached for BankManager role
 }
 }
}
404 {
Class public ClientClassB {
 Method B {
 try {
 ServerClassM.M(); // Might throw an authorization exception.
 // Try block fails to execute when invoked
 // by entity with improper authorization,
 // such as BankTeller
 }
 catch (AuthException e)
 {
 // Codepath reached for BankTeller role
 }
 }
}
408 — Class public ServerClassM {
 Method M () enforces ManagerAuthorization.AuthMethod
}
406 {
Class public ManagerAuthorization {
 Method AuthMethod () throws AuthException {
 // Body of AuthMethod containing authorization process
 }
}

FIG. 4A

System and process for enhancing method calls of special purpose object-oriented
programming languages to have security attributes for access control

5/10

402 {
Class public ClientClassA {
 Method A {
 try {
 ServerClassM.M(); // Might throw an authorization exception.
 // Try block successfully executes when
 // invoked by entity with proper authority,
 // such as BankManager role
 }
 catch (AuthException e)
 {
 // Codepath not reached for BankManager role
 }
 }
}
404 {
Class public ClientClassB {
 Method B {
 try {
 ServerClassM.M(); // Might throw an authorization exception.
 // Try block fails to execute when invoked
 // by entity with improper authorization,
 // such as BankTeller role
 }
 catch (AuthException e)
 {
 // Codepath reached for BankTeller role
 }
 }
}
410 {
Class public ServerClassM enforces ManagerAuthorization.AuthMethod {
 Method M () {
 } 412
 }
Class public Manager Authorization {
 Method AuthMethod () throws AuthException {
 }
}

FIG. 4B

6/10

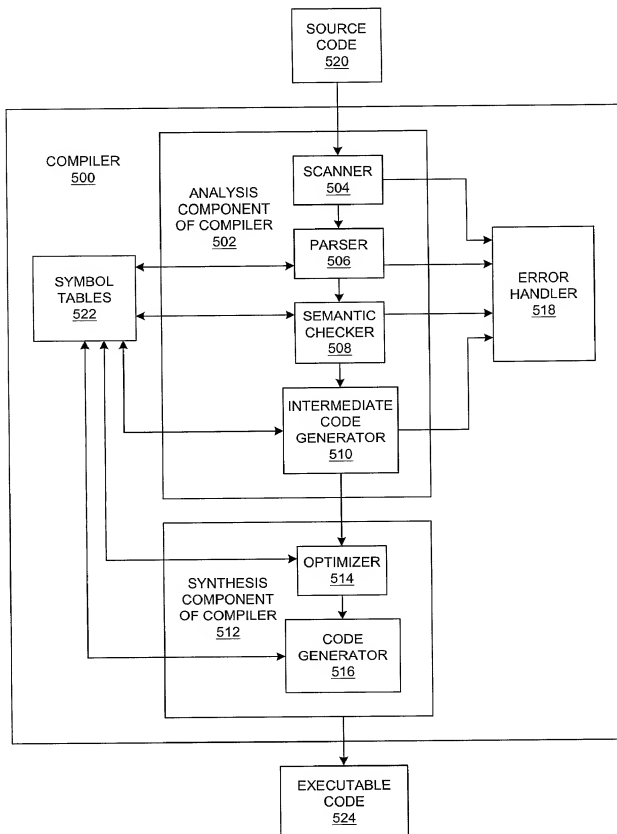


FIG. 5
(PRIOR ART)

System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

7/10

dec → classdec

classdec → **class** class_id **extends** class_id { { class_field } }

classdec → **class** class_id **extends** class_id **enforces** authMethod { { class_field } }

class_field → vardec

class_field → method

602

method → **method** id(tyfields) **enforces** authMethod — 608

method → **method** id(tyfields) = exp

authMethod → **method** id(tyfields) = exp — 604

method → **method** id(tyfields):type_id = exp

authMethod → **method** id(tyfields):type_id = exp — 606

exp → **new** class_id

exp → lvalue id()

exp → lvalue id(exp {, exp})

FIG. 6

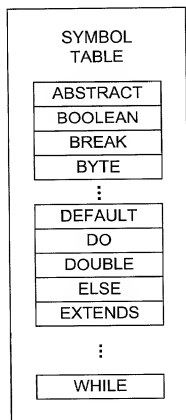


FIG. 7A

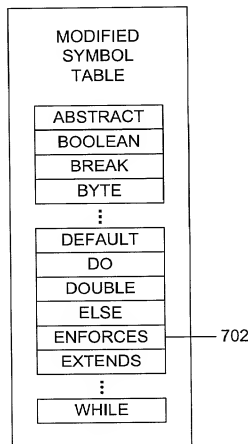


FIG. 7B

System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

8/10

```
802 — class CC extends Object {  
    method A ( ) = { SC:M ( ) }  
}  
  
804 — class SC extends Object enforces ACLClass:R {  
    method M ( ) = { body of M }  
    method F ( ) = { body of F }  
}  
  
806 — class SC extends Object {  
    method M ( ) enforces ACLClass:R = { body of M }  
    method F ( ) = { body of F }  
}  
  
808 — class SCSC extends SC {  
    method M ( ) = { body of M }  
}
```

FIG. 8

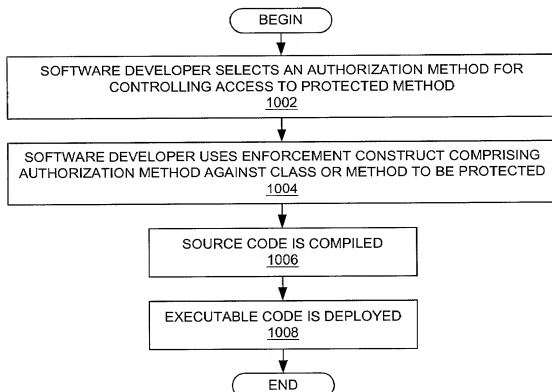


FIG. 10A

System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

9/10

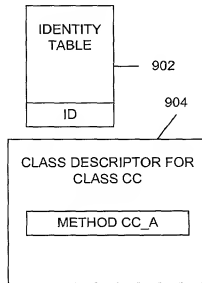


FIG. 9A

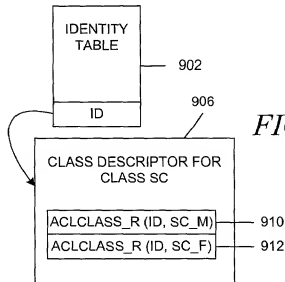


FIG. 9B

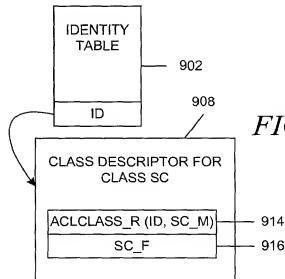


FIG. 9C

System and process for enhancing method calls of special purpose object-oriented programming languages to have security attributes for access control

10/10

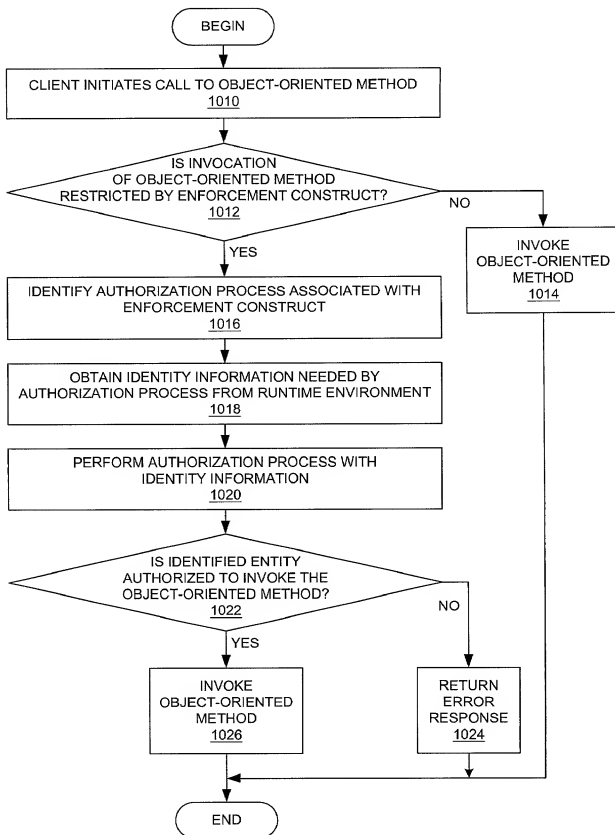


FIG. 10B